

REMARKS

Applicants respectfully traverse and request reconsideration.

Applicants respectfully note that claim 18 has not been addressed in the present Office Action. Applicants therefore submit that claim 18 is in proper condition for allowance.

Claim 17 stands rejected under 35 U.S.C. § 112, ¶ 1 as failing to comply with the enablement requirement. However, the Office Action appears to confuse the written description requirement with the enablement requirement by stating that “there is a lack of written description in the specification which specifically states that the allocating of the program code among the plurality of processors for processing is done without regard to a processor mode.” (Emphasis in original). Applicants submit that the legal standard for determining whether a claim meets the enablement requirement of 35 U.S.C. § 112, ¶ 1 is distinct from that of the written description requirement of the same section and paragraph of U.S. Code. (MPEP § 2164). While the written description requirement determines whether the inventor was in possession of the invention (MPEP § 2163.02), the enablement requirement determines whether one having ordinary skill in the art could make and use the claimed subject matter from the disclosure in the patent application coupled with information known in the art without undue experimentation. (MPEP § 2164.01). Because the rejection is formally introduced as one pertaining to the enablement requirement and because one having ordinary skill in the art could make and use the claimed subject matter from the disclosure in the patent application coupled with information known in the art without undue experimentation, Applicants respectfully request that this rejection be withdrawn.

In the instant matter, claim 17 includes, among other things, a plurality of processors coupled to a bus and kernel program code configured to dynamically allocate the processing of the program code among the plurality of processors without regard to a processor mode.

Applicants respectfully note that throughout the written description, Applicants discuss the x86 processor as one example of a processor. (*See e.g.*, pp. 6–7, 9–10, etc.). As taught in U.S. Patent No. 5,706,514 to Bonola (“Bonola”), CPUs including the x86 family of processors from the Intel Corporation may operate in one of up to at least three different modes: real, protected or virtual. (Cols. 1–2, etc.). Applicants further note that at no point in the instant specification is the mode of a processor relevant, required or otherwise necessarily related to the disclosed subject matter.

Given that one having ordinary skill in the art would recognize that processors such as those disclosed in the instant application may operate in one of a variety of modes (e.g., real, protected or virtual) and given that the originally filed specification did not relate the disclosed subject matter with a processor mode, Applicants respectfully submit that one having ordinary skill in the art could make and use the claimed subject matter based on the disclosure in the patent application coupled with information known in the art without undue experimentation. Restated, although one having ordinary skill in the art would recognize that processors may operate in a given mode (e.g., real, protected or virtual), one would also recognize, based on a reading of the originally filed application, that a kernel program code configured to dynamically allocate the processing of the program code among the plurality of processors may be performed without regard to a processor mode. Applicants note that the absence of a discussion directed toward processor modes is not fatal to the determination of enablement because a patent need not teach, and preferably omits, what is well known in the art. (MPEP § 2164.01). Thus, an apparatus having, among other things, a kernel program code configured to dynamically allocate the processing of the program code among the plurality of processors without regard to a processor mode, is enabled.

Claims 2–12 and 15–16 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Bonola. Claims 15 and 16 have been amended to be more consistent with the written description and preamble. New claims 19–20 include allocating the available processing resources among the tasks based on the capabilities of each of the available processors of the homogeneous multiprocessor environment comprises allocating the available processing resources among the tasks based on the ability of each of the available processors of the homogeneous multiprocessor environment to be divided or aggregated with another processor of the homogeneous multiprocessor environment to provide a processing resource. Applicants submit that the originally-filed application contains support for the subject matter presented in new claims 19–20 at least on page 8. Although not addressed in the present Office Action, claim 18 has been amended similar to claims 15–16. Additionally, claim 21 has been added to mirror the limitations presented in new claims 19–20.

Claims 15–16 and 18 have further been amended to indicate that the available processing resources in the homogeneous multiprocessor environment are identified independent of the tasks.

Bonola is directed toward the distributed execution of mode mismatched commands in multiprocessor computer systems. Bonola explains that processors such as the x86 and Pentium microprocessors may operate in different processor modes (e.g., real, protected and virtual). (Col. 1). As taught, a processor operating in one mode may not be able to execute code written for a different mode. For example, “protected mode still does not support MS-DOS ...[.] therefore, code executing in protected mode cannot use most BIOS level service routines and MS-DOS software interrupts and must rely on slow software emulation when such services are required or utilized lengthy reflection techniques.” (Col. 2, ll. 19–28). It is further taught that

transitions between modes may: disable features of the previous mode, disrupt access to memory, and disable multitasking protections mechanisms leading to undesirable thread suspension or termination or data corruption. Additionally, critical steps and clock cycles are often required in any such transition. (Col. 2, ll. 53–65). Thus, Bonola states that “it would be desirable to utilize features and system resources accessible to a particular microprocessor mode without resorting to slow software emulation or altering the current mode of the 80386 system microprocessor.” (Col. 2, l. 66 to Col. 3, l. 2).

According to the Bonola disclosure, this desire is satisfied by the addition of at least one microprocessor (the slave) connected to the 80386 processor (the host) such that the slave offloads software interrupt handling and other mode specific commands encountered by the host microprocessor when the host is not in the right mode to execute the instructions. (Col. 3, ll. 5–30). Conspicuously absent from at least the cited portions of Bonola is Applicants’ claimed “identifying available processing resources in the homogeneous multiprocessor environment independent of the task.” Additionally, Applicants are further unable to find any teaching or suggestion in the cited portions of Bonola where tasks are performed and wherein the functional programs cause the available processing resources to perform the tasks of at least one of: graphics imaging processing, video processing, audio processing and communications processing.

As to claim 15, the present Office Action cites column 7, lines 36–38 as teaching Applicants’ claimed identifying available processing resources in the homogeneous multiprocessor environment. However, Applicants respectfully note that column 7, lines 36–38 are silent as to Applicants’ claimed “identifying available processing resources in the homogeneous multiprocessor environment independent of the task.” (Emphasis added). In fact,

Yamamoto appears to teach the opposite where he states that “At step 1011, a determination is made whether a slave CPU 21–23 is available to offload execution of the mode mismatched command. Preferably, this be will [sic] accomplished by polling each slave CPU’s multiprocessor interrupt ports for its current mode and activity state described in FIGS. 1 and 2 above. Alternatively, the host CPU 20 could look up the current status of all slave CPUs 21–23 present in the system by means of a lookup table held in a processor status memory Each slave CPU will be responsible for maintaining its own status entries, including configuration mode (e.g., protected V86, or real).” (Emphasis added, col. 7, ll. 35–50). In other words, in each embodiment, Yamamoto appears to teach that the alleged determination is dependent upon the command being mismatched. Because Applicants’ claim identifying available processing resources independent of the task, it is respectfully submitted that claim 15 is in proper condition for allowance.

The Office Action further cites column 1, lines 15–26 of Bonola as teaching an X86 system. The Office Action alleges that Applicants; claimed “performing the tasks ... wherein the functional programs cause the available processing resources to perform the tasks of at least one of: graphics image processing, video processing, audio processing and communications processing” because these types of processing “are just some of the types of tasks that can be performed on an X86 system.” Applicants respectfully submit that the cited portion of Bonola fails to teach or suggest any type of graphics image processing, video processing, audio processing and communication processing. Instead, this portion of Bonola merely suggests that many popular personal computers are based on the Intel 8086 family of microprocessors. These “complex instruction set microprocessors have been designed for upward compatibility.” Applicants are unable to find any teaching within this portion of Bonola that suggests this feature

of the claim. For this reason alone, Applicants respectfully believe claim 15 to be allowable over the cited prior art.

Moreover, Applicants respectfully note that the Office Action's characterization of Bonola ignores claim language and is contrary to Applicants' disclosure. For example, the Office Action's contention improperly ignores claim language when it states that these types of processing "are just some of the types of tasks that can be performed on an x86 system." (Emphasis added). As best understood, this statement indicates that graphics image processing, video processing, audio processing and/or communications processing may be performed on a system and not specifically a processor or processing resource. It is Applicants' understanding that a system generally includes, among other things, a processor and other application-specific circuits. In other words, this statement mirrors Applicants' description of the prior art (as provided below) where additional application-specific circuits are needed in conjunction with the x86 microprocessor to process multimedia data. (*See e.g.*, p. 9). Because the Office Action does not appear to address claim language, claim 15 is believed to be in proper condition for allowance.

Additionally, Applicants note that the Office Action's assertion runs afoul of Applicants' written description. Applicants respectfully draw the Examiner's attention to, among other places, page 9 of Applicants' originally filed application where it is taught that:

In a traditional computer system, applications 603 are executed on an operating system 602, which, in turn, is executed on an x86 microprocessor 601. The internal architecture of the x86 microprocessor evolved before the processing of multimedia data and, therefore, does not provide substantial native support for input/output capability as is needed for the processing of multimedia data.

Moreover, the internal architecture of the x86 microprocessor is not optimized for efficient processing of multimedia data. Therefore, additional application-specific circuits are needed in connection with the x86 microprocessor Thus, the x86 microprocessor and the application-specific circuits are immutably configured to

perform their pre-determined roles, their processing capabilities cannot be adjusted or re-allocated.

Applicants further note that graphics image processing, video processing, audio processing and communications processing are generally-accepted examples of multimedia processing (i.e., the processing of multimedia data). In other words, it is Applicants' understanding that the above-listed and claimed types of processing are not "just some of the types of tasks that can be performed on an X86 system" for at least the reason that the internal architecture of the x86 microprocessor evolved before the processing of multimedia data, as explained above, and thus the x86 microprocessor is generally incapable of performing these types of processing. For at least this reason, claim 15 is believed to be properly allowable over Bonola. If the Examiner maintains the instant rejection, Applicants respectfully ask the Examiner to produce a specific citation indicating by column/page and line number where these types of tasks may be performed using an X86 processor.

Claims 16 and 18 contain similar limitations as argued above with respect to claim 15. Accordingly, Applicants respectfully reassert the relevant remarks made above with respect to claim 15. For at least these reasons, claims 16 and 18 are also believed to be in proper condition for allowance.

Claims 2–12 depend upon allowable claim 15 and further add additional novel and non-obvious subject matter. For instance and with respect to claim 3, Applicants claim a method wherein a plurality of processors ... are capable of executing a first instruction of a first instruction set and a second instruction of a second instruction set wherein the first instruction and the second instruction share an identical bit pattern but perform different operations. The Office Action cites column 1, lines 15–26 of Bonola as allegedly teaching this feature. Applicants, however, note that the cited portion of Bonola merely teaches that many popular

personal computers are based on the Intel 8086 family of microprocessors and that these “complex instruction set microprocessors have been designed for upward compatibility.” In other words, the cited portion of Bonola does not appear to teach or suggest the claimed features. Accordingly, Applicants respectfully believe that claims 2–12 are in condition for allowance.

As to claims 19–21, Applicants respectfully reassert the relevant remarks made above with respect to claims 15–16 and 18 and further note that claims 19–21 not only depend upon allowable independent claims but further add additional novel and non-obvious subject matter. While the Office Action cites column 7, lines 42–52 and column 8, lines 11–13 of Bonola as allegedly reaching this claim feature, Applicants respectfully submit that these portions of Bonola merely teach: identifying available processing resources based on a configuration mode (i.e., a processor mode of real, protected or virtual) and activity state (e.g., working or idle); and loading parameters necessary to successfully carry out a mode mismatched command on the selected slave processor. Thus, neither portion of Bonola teaches or suggests Applicants’ claimed subject matter directed to allocating the available processing resources among the tasks based on the ability of each of the available processors of the homogeneous multiprocessor environment to be divided or aggregated with another processor of the homogeneous multiprocessor environment to provide a processing resource. For at least these reasons, claims 19–21 are further believed to be allowable over the cited prior art.

As to claim 17, the present Office Action states that:

Bonola discloses ‘a technique for handling processor mode mismatched instructions or commands encountered by a CPU within multiprocessor computer system ... if a multimode processor encounters a command or instruction that it cannot execute without shifting modes or mode emulation, it will look for an alternate processor present in the computer system to instead handle the mode mismatched command’ In other words, Bonola’s system can perform the resource allocation among the processors regardless of a processor mode.

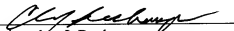
Based on this teaching, the Office Action then asserts that Bonola allegedly suggests the dynamic allocation of processing of the program code among the plurality of processors without regard to a processing mode. Applicants respectfully note that the Office Action correctly quotes from the Bonola reference but then mischaracterizes the teaching in an attempt to read on Applicants' claim language. As cited, Bonola's system first determines if it can process a particular command or instruction based on the current processor mode. If the multimode processor cannot execute the command or instruction without shifting modes or mode emulation, then it will look to an alternate processor that can handle the mode mismatched command (based on the alternate processor's current processor mode). Thus, it is clear that Bonola's system always considers the processor mode to determine what processor within a multiprocessor computer system should execute the command or instruction. Applicants respectfully submit that it is irrelevant that Bonola's system may always execute the command or instruction using a given processor because in each circumstance, that processor's mode must be properly matched before execution.

Because Bonola teaches a system where a knowledge of a processor mode is required in order to determine which processor of a multiprocessor system should execute a command or instruction, Bonola does not appear to teach or suggest Applicants' claimed kernel program code configured to dynamically allocate the processing of the program code among the plurality of processors without regard to a processor mode. For at least this reason, claim 17 is believed to be in proper condition over the cited prior art.

In light of the foregoing, Applicants respectfully submit that the present application is in condition for allowance and respectfully request that a Notice of Allowance be issued in this case.

Respectfully submitted,

Date: 9/12/06

By: 
Christopher J. Reckamp
Registration No. 34,414

Vedder, Price, Kaufman & Kammholz, P.C.
222 N. LaSalle Street
Chicago, IL 60601
(312) 609-7500
FAX: (312) 609-5005